RESEARCH ARTICLE

Knowledge relation rank enhanced heterogeneous learning interaction modeling for neural graph forgetting knowledge tracing

Linqing Li[1], **Zhifeng Wang**[1,2]*

1 Central China Normal University Wollongong Joint Institute, Central China Normal University, Wuhan, China, **2** Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, China

* zfwang@ccnu.edu.cn

Citation: Li L, Wang Z (2023) Knowledge relation rank enhanced heterogeneous learning interaction modeling for neural graph forgetting knowledge tracing. PLoS ONE 18(12): e0295808. https://doi.org/10.1371/journal.pone.0295808

Editor: Ayesha Maqbool, National University of Sciences and Technology NUST, PAKISTAN

Received: August 24, 2023

Accepted: November 29, 2023

Published: December 22, 2023

Data Availability Statement: All relevant data are within the paper.

Funding: This study was funded by the National Natural Science Foundation of China [62177022, 61901165, 61501199], the AI and Faculty Empowerment Pilot Project [CCNUAI&FE2022-03-01], the Collaborative Innovation Center for Informatization and Balanced Development of K-12 Education by MOE and Hubei Province [xtzd2021-005], and the Natural Science Foundation of Hubei Province [2022CFA007] in the form of grants to ZW.

Abstract

Knowledge tracing models have gained prominence in educational data mining, with applications like the Self-Attention Knowledge Tracing model, which captures the exercise-knowledge relationship. However, conventional knowledge tracing models focus solely on static question-knowledge and knowledge-knowledge relationships, treating them with equal significance. This simplistic approach often succumbs to subjective labeling bias and lacks the depth to capture nuanced exercise-knowledge connections. In this study, we propose a novel knowledge tracing model called Knowledge Relation Rank Enhanced Heterogeneous Learning Interaction Modeling for Neural Graph Forgetting Knowledge Tracing. Our model mitigates the impact of subjective labeling by fine-tuning the skill relation matrix and Q-matrix. Additionally, we employ Graph Convolutional Networks (GCNs) to capture intricate interactions between students, exercises, and skills. Specifically, the Knowledge Relation Importance Rank Calibration method is employed to generate the skill relation matrix and Q-matrix. These calibrated matrices, alongside heterogeneous interactions, serve as input for the GCN to compute exercise and skill embeddings. Subsequently, exercise embeddings, skill embeddings, item difficulty, and contingency tables collectively contribute to an exercise relation matrix, which is then fed into an attention mechanism for predictions. Experimental evaluations on two publicly available educational datasets demonstrate the superiority of our proposed model over baseline models, evidenced by enhanced performance across three evaluation metrics.

Introduction

In the age of rapid technological advancement, the ubiquity of networks has ushered in a wealth of information and convenience. This surge in data, commonly referred to as big data, plays a pivotal role in interconnecting diverse sectors such as education, healthcare, and transportation. The fusion of education with information science is an inexorable trend, catalyzing an expansive realm of research in online education technologies. Notably, the application of artificial intelligence and related technologies to analyze substantial educational data yields invaluable insights and enhances the educational landscape [1–4].

Competing interests: The authors have declared that no competing interests exist.

Online educational platforms, like edX, Coursera, and Udacity, have gained widespread adoption for tracking, reporting, and delivering online courses [3, 5, 6]. These platforms offer students a spectrum of advantages, including diverse online courses and personalized learning resources [6–8]. Moreover, they provide a valuable alternative to traditional classroom teaching, circumventing geographical or weather-related limitations. Teachers leverage these platforms to tailor remedial materials to suit individual student needs [4, 8–11].

Central to the effectiveness of online education is the task of tracking students' performance based on their past interactions [12, 13]. This task, known as *Knowledge Tracing* [14, 15], seeks to gauge students' knowledge states by analyzing their responses to exercises. Essentially, Knowledge Tracing (KT) assesses students' mastery of knowledge. Given a set of practice questions $X_t = (x_1, x_2, \ldots, x_t)$ and response logs (e.g., correct or incorrect), KT predicts the likelihood of correctly anticipating students' knowledge states in subsequent interactions $(p(r_{t+1} = 1|e_{t+1}, X))$ based on past interactions and responses. Here, x_t is presented as the tuple (q_t, a_t), where q_t represents the student's question and a_t denotes the corresponding response. Refer to Fig 1 for a visual representation of the knowledge tracing model's framework.

Numerous knowledge tracing models have emerged to tackle the challenge of monitoring students' knowledge states, such as Deep Knowledge Tracing (DKT) [16], Dynamic Key-Value Memory Network (DKVMN) [17], Graph-based Knowledge Tracing (GKT) [18], and Self-Attentive Model for Knowledge Tracing (SAKT) [19]. These models exhibit improved predictive performance across several educational datasets. DKT incorporates deep neural networks to forecast student performance. However, it overlooks knowledge point information and student abilities. DKVMN employs nonlinear transformations to learn representations and master levels for each concept, but disregards the similarity between knowledge concepts (KCs) when making predictions. SAKT delves into the KC-exercise relationship for mastery level prediction but ignores the potential organizational structure of past interactions as a graph.

To address these limitations, we present the Knowledge Relation Rank Enhanced Heterogeneous Learning Interaction Modeling for Neural Graph Forgetting Knowledge Tracing (NGFKT). This model rectifies the shortcomings of existing models when predicting student performance from past interactions. The model encompasses four key steps: (1) Skill relation matrix and Q-matrix calibration using the Knowledge Relation Importance Rank Calibration method (KRIRC), which considers the students-exercise-KCs relationship; (2) Inputting the calibrated matrices into a Graph Convolutional Network (GCN) to yield skill and exercise embeddings; (3) Incorporating heterogeneous interactions, item difficulty, skill embeddings, and exercise embeddings to generate an exercise relation matrix; (4) Applying the Position-Relation-Forgetting attention mechanism, inspired by student forgetting behavior, to predict student performance. We validate our model on two datasets, demonstrating its superiority over traditional models.

The contributions of this paper encompass:

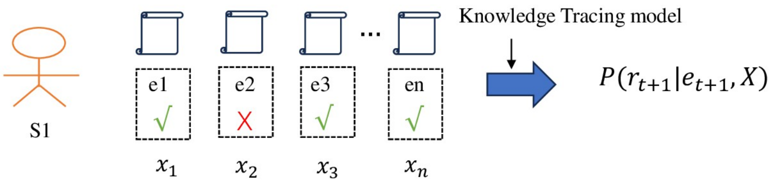

Fig 1. Illustration of the knowledge tracing model's process.

https://doi.org/10.1371/journal.pone.0295808.g001

1. **Calibration Method for Enhanced Skill Relation Matrix and Q-Matrix**: By considering the intricate relationship between knowledge concepts (KCs) within heterogeneous interactions—namely students, exercises, and KCs—the calibration method ensures a more accurate representation. This calibrated skill relation matrix and Q-matrix, coupled with the heterogeneous interactions, serve as inputs for a Graph Convolutional Network (GCN). This integration facilitates the generation of comprehensive exercise embeddings and skill embeddings, effectively capturing the intricate interplay between students, exercises, and KCs.

2. **Exercise Relation Modeling with Enhanced Forgetting Mechanism**: In contrast to the GKT model [18], our proposed approach goes beyond surface-level prediction by incorporating the nuanced forgetting behavior of students and relative distance representations. This enhancement is realized through the utilization of the Position-Relation-Forgetting attention mechanism. This attention mechanism is designed to predict student performance, and its integration demonstrates the model's ability to capture deeper insights into learning dynamics.

3. **Comprehensive Experimental Evaluation**: Our evaluation strategy addresses three key aspects. First, we measure the predictive efficacy of the NGFKT model against baseline models using three established metrics. Second, we assess the effectiveness of the NGFKT model even when dealing with limited records. Finally, we employ radar diagrams to visually depict and understand the knowledge tracing outcomes. This meticulous experimental evaluation bolsters the robustness of our proposed model and showcases its superiority in various scenarios.

The rest of this paper unfolds as follows: The next section, "Related Works," delves into knowledge tracing models, graph neural networks, relation modeling, and attention mechanism. Subsequently, the "Methods" section provides a detailed exposition of NGFKT's structure. The "Experiments" section presents implementation details and experimental outcomes. Finally, we conclude and highlight avenues for future research in the "Conclusions and Future Work" section.

Related works

Knowledge tracing

Knowledge Tracing (KT) aims to gauge students' comprehension levels based on their historical interactions. The effectiveness of deep learning techniques in fields like speech processing (e.g., [20–23]) and computer vision (e.g., [24–27]) has inspired the development of deep learning-based KT models. One pioneering model is the Deep Knowledge Tracing (DKT) [16], which employs neural networks to capture intricate educational processes. The DKT+ model [28] extends DKT by introducing regularization items. However, these approaches overlook the influence of exercise embeddings. The Enhanced Knowledge Tracing (EKT) model [29] utilizes exercise embedding modules for predicting students' future performance. The Memory Augmented Neural Network (MANN) [30] employs Key and Value metrics to model exercise embeddings and make predictions. Yet, these methods often neglect relation modeling to predict students' knowledge state. To address this gap, the Self-Attentive Knowledge Tracing (SAKT) model [19] explores relationships between students, exercises, and skills in past interactions. Additionally, knowledge tracing can be framed as a graph-based task. The Graph-based Knowledge Tracing (GKT) model [18] formulates knowledge tracing as a time series node-level classification problem within a graph. However, the GKT model does not take exercise relation modeling and the heterogeneous interactions into consideration. Therefore, the our model is proposed to handle these limitations and further improve the prediction performance of the knowledge tracing model.

Graph neural networks

Graph structures, which can represent connections and entities in the real world, are more intricate than simple tree structures [31]. Graph Convolutional Networks (GCNs), a unique type of neural network, operate directly on graph-structured data. A GCN updates node representations by considering the nodes themselves and their neighbors [32]. By incorporating multiple graph convolutional layers, GCNs ensure that updated nodes accurately capture both higher-order neighbor features and neighboring node features.

Several knowledge tracing models explore the utility of knowledge graph structures. Chanaa et al. [33] employ a dynamic graph structure in which nodes represent students and edges evolve over time. However, this method overlooks the heterogeneous interactions among students, exercises, and knowledge concepts (KCs). The Graph-Infused Knowledge Tracing (GIKT) model [13] treats the exercise-skill relationship graph as a bipartite graph and utilizes embedding propagation within GCNs to incorporate exercise-skill correlations. Nevertheless, GIKT doesn't fully account for student behavior. Our proposed Neural Graph Forgetting Knowledge Tracing (NGFKT) model uses skill relation matrices and Q-matrices to model exercise-KC relationships, employing an attention mechanism enhanced with a forgetting layer to predict performance.

Relation modeling

Relation modeling, including exercise relation modeling, is crucial in knowledge tracing tasks [34]. When two exercises are connected through shared knowledge concepts or students, their relation becomes significant [35]. Additionally, the Q-matrix can reveal relationships between exercises and skills, unveiling connections between exercise pairs [36]. Intuitively, if two problems share similar difficulty and past practice, they can be considered related [37, 38]. Building on these concepts, NGFKT integrates item difficulty, exercise embedding, skill embedding, and a contingency table to formulate a comprehensive exercise relation matrix.

Attention mechanism

The attention mechanism is a powerful tool for sequence modeling tasks [39–41]. It enhances predictions by focusing on crucial input elements. The mechanism determines attention weights for input vectors, enabling it to extract relevant words in machine translation tasks [42–46]. In our model, we introduce a novel attention mechanism to predict student performance. This mechanism incorporates exercise relation modeling along with a forgetting behavior component, further enhancing prediction accuracy.

Motivation

This paper aims to provide an innovative knowledge tracing model: NGFKT to estimate the student knowledge state based on the heterogeneous interactions between students, exercises, and skills. Specifically, the first motivation is to design a calibration method to generate the Q matrix and skill relation matrix. The second motivation is to develop an exercise relation modeling method to obtain an exercise relation matrix. Finally, incorporating the attention mechanism with the relation matrix predicts student performance.

Table 1. Important notations.

Notations	Description
KT	The knowledge tracing model
KC	Knowledge concepts
$\hat{Q}$	The calibrated Q-matrix
$\hat{S}$	The calibrated skill reltion matrix
M	The calibrated matrix
a_{rank}	The knowledge relation importance rank of the element: "a"
b_{rank}	The knowledge relation importance rank of the element: "b"
K	The number of the knowledge concepts
N	The number of the questions
σ	The standard deviation
$\hat{e}$	Skill-exercise embedding
q	One question in the question set
s	One student in the student set
$\psi_{s,q,t}$	the cognitive difficulty of the question: q for the student: s
R^E	The exercise relation matrix
I_i	The input element of the Position-Relation-Forgetting attention mechanism

https://doi.org/10.1371/journal.pone.0295808.t001

Preliminaries

In this section, on the one hand, we summarize the important mathematical notation covered in this paper in Table 1, and on the other hand, we give important terminology and problem definitions.

Terminologies

Heterogeneous interactions. There are numerous interactions between students, exercises, and skills referring to the Fig 2. Heterogeneous interaction (H) is formed up of an object set, V, and a link set, E. Students, exercises, and skills are all object types in V. E is a set of relational types of the type E = (r_A, r_C). The r_A indicates the response of exercises and r_C means the relation it involved.

Knowledge tracing. Knowledge Tracing (KT) evaluates learners' knowledge mastery. KT predicts the likelihood of effectively estimating students' knowledge states in coming meets given a set of test questions and answer logs (e.g., correct or incorrect).

Student modeling. At the same time, based on the results of knowledge tracking, we can model student groups and individual students. Specifically, if the input is a student group, then the result of our knowledge tracking is the modeling of the student group. If the input is an individual student, then the knowledge tracking is the modeling of a single student.

Exercise modeling. In this paper, we model the exercises based on the exercise relationship on the heterogeneous interaction graph. Specifically, we apply the graph neural network to process the exercises and discuss the difficulty of the exercises to generate the final exercise embedding.

Knowledge modeling. In order to comprehensively model the knowledge in this paper, we apply the knowledge relation importance rank calibration method to model the knowledge contained in the exercises and generate the skill relation matrix. Then this matrix is treated as the input of GCN to generate the hidden embedding of exercises.

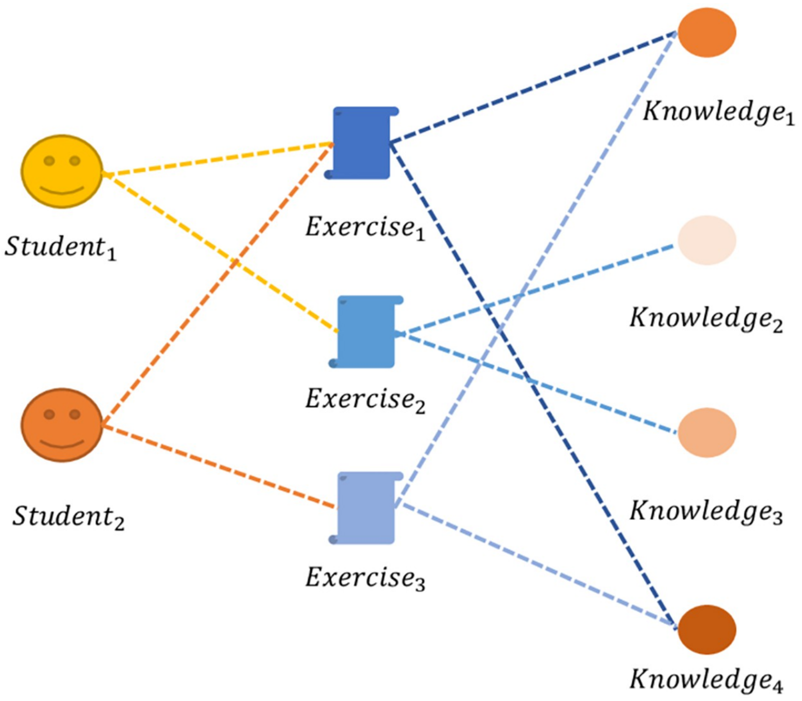

Fig 2. The heterogeneous interaction between student, exercise, and skills.

https://doi.org/10.1371/journal.pone.0295808.g002

Problem definition

The computerized system generates the student's responding records given a question set of n exercises (e.g., e_0, e_1, e_2. . .e_n) for the student in the smart educational system from Timestamps 1 to t. Those interactions are designated as S = (s_1, s_2, s_3. . ... s_n), and each interaction s_i is presented as a tuple: $s_i = (e_i, r_i, t_i)$, where e_i is the exercise that this student attempted, $r_i \in \{0, 1\}$ is the student's answer, and t_i is the time at which s_i happens. Our goal is to estimate the knowledge state of students based on previous information. Specifically, we input the heterogeneous interaction graph and previous student records: S, the output of the knowledge tracing model is the knowledge state of students.

Methods

In this section, the NGFKT model is developed based on the exercise relation matrix and Position-Relation-Forgetting attention mechanism to predict the performance of students. There exist three steps including the skill relation matrix and Q-matrix modeling, the extraction of the exercise relation matrix, and the predictions based on the Position-Relation-Forgetting attention mechanism. First, the skill relation matrix($\hat{S}$) and the calibrated Q-matrix ($\hat{Q}$) are designed based on the KRIRC method and serve as the input of the model. Then, the skill relation matrix $\hat{S}$, and the calibrated Q-matrix ($\hat{Q}$) are treated as the inputs of the GCN to output the embedding of exercises and skills in the heterogeneous interactions. The similarity of exercises can be computed based on the exercise embedding, skill embedding, item difficulty, and contingency table to generate the corresponding exercise relation matrix(R^E). Finally, the exercise relation matrix(R^E) serves as the input of the Position-relation-Forgetting attention mechanism to track the student knowledge state. The overall structure can refer to Fig 3.

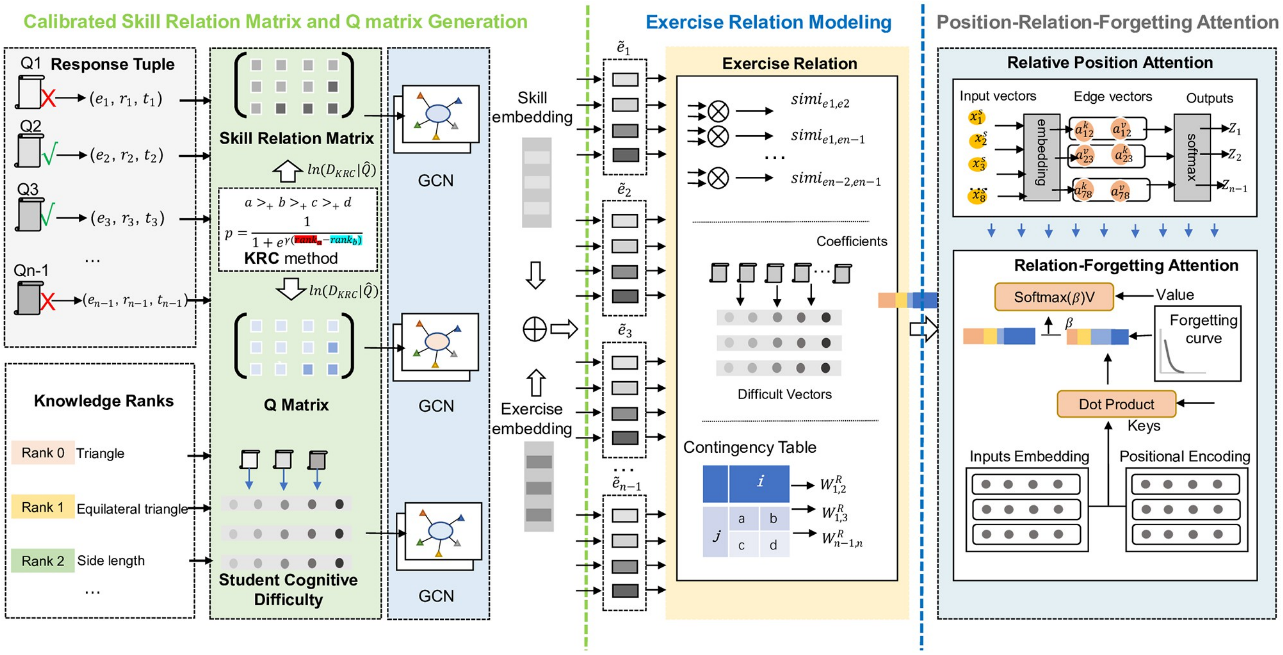

Fig 3. The overall structure of the Neural Graph Forgetting Knowledge Tracing (NGFKT). Firstly, according to previous interactions, the past interactions of students and knowledge levels can be extracted from the inputs. Then KRIRC method is designed to generate the calibrated skill relation matrix and Q-matrix. The skill relation matrix, Q-matrix, and the cognitive difficulty of each student are treated as the inputs of the GCN to generate the skill-exercise embedding: $\hat{e}$. Secondly, the $\hat{e}$, the item difficulty of each question, and the contingency table are combined to output the exercise relation matrix. Finally, the Position-Relation-Forgetting attention mechanism is utilized to process the inputs to make predictions.

https://doi.org/10.1371/journal.pone.0295808.g003

Skill relation matrix and Q matrix modeling

We design this part to extract the heterogeneous information between students, exercises, and skills. The basic idea of this part is that the related skill can be considered as the skill that is covered by the same exercise. Considering the hierarchical knowledge levels of skill, the related skill also can be defined as the parent nodes and child nodes of the skill. For instance, the knowledge concept: the "Triangle" is viewed as the "Right Triangle"'s parent node, and the "Pythagorean Theorem" is the child node of the "Right Triangle". Therefore, the related skills of the "Right Angle" are the concepts: "Triangle" and "Pythagorean Theorem". Inspired by these two ideas, the importance of the skills is ranked based on the following partial order, and the calibrated method called the knowledge Relation Importance Rank Calibration method is proposed(KRIRC). A pairwise Bayesian treatment is as follows. For convenience, we define a partial order $>^+_i$ as:

$$a >^+_i b >^+_i > c >^+_i d \tag{1}$$

where "a", "b", "c", and "d" are the neighbors of the skill. Here, "a" implies the skill of the parent node, knowledge level is 0, in the knowledge level graph. "b" denotes the skill of the child node, knowledge level is 1, in the knowledge level graph. "c" can be interpreted in a similar way. "d" is the skill that is covered by the same exercise. Along this line, neighbor: "a" is more important than neighbor: "b" in extracting the neighbor of the skill. The rank of the skill: "a", "b", "c", and "d" are 0, 1, 2, and 3 respectively. Thereby, according to the Eq 1, the partial order relationship set can be defined as $D_{KRIRC} = \{(i, a, b)|a >^+_i b, i = 1, 2, 3 \ldots K\}$ where K is the number of knowledge concepts. Based on the traditional Bayesian method, we assume the calibration matrix: $\hat{M}$ such as $\hat{Q}$ or $\hat{S}$ uniforms the Gaussian distribution with the standard deviation of each dimension. To give the calibration matrix labels higher confidence, we define $p(a >_i^+ b|\hat{M})$ as follows:

$$p(a >_i^+ b|\hat{M}) = \frac{1}{1 + e^{\lambda(a_{rank} - b_{rank})}} \tag{2}$$

where λ controls the discrimination of relevance values of different knowledge levels. And the a_{rank} and b_{rank} means the importance of the element a or b. The log posterior over D_{KRIRC} on $\hat{M}$ can be eventually computed as:

$$\begin{aligned} lnp(\hat{M}|D_{KRIRC}) &= ln \prod_{(i,a,b)\in D_{KRIRC}}^{1} p(a >_i^+ b|\hat{M}_i)p(\hat{M}_i) \\ &= \sum_{i=1}^{N}\sum_{a=1}^{K}\sum_{b=1}^{K} I(a >_i^+ b) ln \frac{1}{1 + e^{\lambda(a_{rank} - b_{rank})}} + C - \sum_{i=1}^{N}\sum_{j=1}^{K} \frac{\hat{M}_{ij}^2}{2\sigma^2} \end{aligned} \tag{3}$$

where I(*) is the judgment function when the function's condition is met and the function output 1. The σ is the standard deviation in Eq 2 and C, the constant variable, can be ignored to train the matrix. Finally, a calibrated matrix $\hat{M}$ estimated by the KRIRC can be calculated. The Q-matrix and the skill relation matrix can apply the KRIRC method to obtain the calibrated Q-matrix and the calibrated skill relation to reflect the hierarchical knowledge levels between different knowledge points. The specific algorithm can be referenced as follows:

Algorithm 1: Knowledge Relation Importance Rank Calibration Method.

```
Input: Students' historical response dataset: $D = s_1, s_2, ...s_N, s_i = (e_i, s_i, t_i)$;
       The knowledge level graph G; The heterogeneous relation graph: $\tau$;
       Task-learning rate: $\alpha$; Hyper-parameter λ;
Output: The calibrated Q matrix: $\hat{Q}$; The calibrated skill relation matrix: $\hat{S}$.
initialization learning rate $\alpha$ and hyper-parameter λ randomly
while element in G and $\tau$ do
    Extract hierarchical knowledge levels and related skills of each element based on G and $\tau$
    Generate the corresponding knowledge rank using Eq 1
end
while not converged do
    while element in $\hat{Q}$ and $\hat{S}$ do
        Evaluate calibrated skill element $q_{ij}$ using Eq 2;
        Replace the element in $\hat{Q}$ and $\hat{S}$ with a calibrated element
    end
    Update parameters learning rate $\alpha$ and hyper-parameter λ
end
```

Exercise relation modeling

The exercise matrix modeling is designed based on two processes. Firstly, skill-exercise embedding:$\hat{e}$ is calculated by applying the GCN. Secondly, the item difficulty is modeled to estimate the difficulty of each question. And $\hat{e}$, item difficulty, and the contingency table are incorporated to generate the final exercise relation matrix:R^E.

The basic idea of the GCN model is to extract the neighbor information to generate the target node embedding. For the GCN model, we apply it to process the relationship of exercises and skills to generate the hidden exercise embedding. In this case, the heterogeneous information can be extracted from the skill relation matrix and the Q-matrix based on previous matrix modeling. Then, based on the knowledge tracing results, we can model the student states to explore the student information in the heterogeneous interactions. These matrices are served as the input of GCN. The GCN model consists of numerous convolutional layers, and each layer can be updated by the states of itself and the neighbors of nodes. The *i*th node in the graph donated as $node_i$ indicating the skill state s_i or exercise state e_i. The neighboring nodes of $node_i$ are denoted as the node set: Node(i). As a result, the ι th layer of the graph convolutional network can be updated as follows:

$$node_i^{\iota} = RELU(\sum_{j \in i \cup Node(i)} w_i^{\iota} node_j^{\iota-1} + b_i^{\iota}) \tag{4}$$

where w^{ι} and b^{ι} infer as the weighted matrix and bias of the GCN layer and the RELU() indicates the activation function accepted in the GCN model. The output of the GCN model:$\hat{e}$ is used for estimating the implicit relations among questions by calculating the inner product of questions:

$$simi_{i,j} = \frac{\hat{e}_i \cdot \hat{e}_j}{|\hat{e}_i||\hat{e}_j|} \tag{5}$$

Then, in order to evaluate the exercises' similarity, we further incorporate the item difficulty with the previous students' performance to generate an exercise relation matrix: R^E. The basic idea is that we apply the student incorrect interactions to estimate the item difficulty and seven metrics, which are used to measure the performance of different variables, to estimate the exercises relation.

For the item difficulty modeling, the students' incorrect interactions can intuitively represent the item difficulty of the exercises involved in the student interactions. Specifically, the students repeat the same questions by utilizing their skills in different timestamps, which can demonstrate the cognitive difficulty of these exercises. In order to model this situation, the cognitive difficulty for the students: s can be defined as follows:

$$\Psi_{s,q,t} = \begin{cases} \left[\frac{|\{R_s == 0\}|_{0:t}}{|Q_q|_{0:t}} * 4\right] & if |Q_q|_{0:t} \geq 5 \\ 5 & otherwise \end{cases} \tag{6}$$

where $\Psi_{s,q,t}$ indicates each student's cognitive difficulty of the question set at timestamp t. The cognitive difficulty is divided into 5 levels including very hard, hard, medium difficulty, relatively easy, and easy. The number ranging from 0 to 5 is accepted to indicate the corresponding levels. The $|Q|$ denotes the set of questions containing the question q before timestamp t and R_s refers to the student's response to the same questions. A zero in the R_s indicates that the student provides a wrong answer for a question. If a learner attempts to answer a question fewer than five times, the cognitive difficulty of this question is directly quantified into 5. Then according to the different learners' cognitive difficulty of questions, the average cognitive difficulty for different learners on the same question is defined as the item difficulty: $\varphi(q)$ after processing the cognitive difficulty for different learners.

$$\varphi(q) = \frac{1}{|\Psi_{s,q,t}|} \sum_{\Psi(s_i,q,t) \in \Psi(s,q,t)} \Psi(s_i, q, t) \tag{7}$$

Then, according to the item difficulty: φ_q, the similarity of question difficulty: *diff* can be

Table 2. The contingency table for exercise *i* and exercise *j*. The labels: “F” and “T” present the student answering the exercise incorrectly or correctly.

		exercise i		
		F	**T**	**total**
exercise j	F	a	b	a+b
	T	c	d	c+d
	total	a+c	b+d	a+b+c+d

https://doi.org/10.1371/journal.pone.0295808.t002

Table 3. Seven evaluation metrics. These metrics are designed to explore the association between two variables.

Metrics	Formulation
Cohen’s Kappa	$W^{Kappa}_{e_i,e_j} = 2(\mathrm{ad} - \mathrm{bc})/(\mathrm{a} + \mathrm{b})(\mathrm{b} + \mathrm{d}) + (\mathrm{a} + \mathrm{c})(\mathrm{c} + \mathrm{d})$
Adjusted Kappa	$W^{Kappa'}_{e_i,e_j} = 2(\mathrm{ad} - \mathrm{bc})/(\mathrm{a} + \mathrm{c})(\mathrm{c} + \mathrm{d})$
Phi coefficient	$W^{Phi}_{e_i,e_j} = (\mathrm{ad} - \mathrm{bc})/\sqrt{(a+b)(b+d)(a+c)(c+d)}$
Yule coefficient	$W^{Yule}_{e_i,e_j} = (\mathrm{ad} - \mathrm{bc})/(\mathrm{ad} + \mathrm{bc})$
Ochiai coefficient	$W^{Ochiai}_{e_i,e_j} = \mathrm{a}/\sqrt{(a+b)(a+c)}$
Sokal coefficient	$W^{Sokal}_{e_i,e_j} = (\mathrm{a} + \mathrm{d})/\sqrt{(a+b+c+d)} = (\mathrm{a} + \mathrm{d})/\sqrt{(a+b+c+d)}$
Jaccard coefficient	$W^{Jaccard}_{e_i,e_j} = \mathrm{a}/(\mathrm{a} + \mathrm{b} + \mathrm{c})$

https://doi.org/10.1371/journal.pone.0295808.t003

modeled as follows:

$$diff(q_i, q_j) = \frac{1}{1 + \varphi(q_i) - \varphi(q_j)} \tag{8}$$

In order to incorporate the previous interaction, the student’s performance on question pair q_i and q_j is summarized in the contingency table. The students’ correct and incorrect responses are interpreted as the mastery indicators of the questions referring to Table 2. When a question pair appears more than once in the previous student interactions, the latest occurrence is taken into consideration. According to the contingency table, seven evaluation metrics, measuring the association between two variables, are developed to measure the relationship between the question pair: e_i and e_j referring to Table 3. A threshold is imposed to control the sparsity of relations of exercises. The output of the contingency table is denoted as W^R, $R \in \{SK, Kappa, Kappa', Phi, Yule, Ochiai, Sokal, Jaccard.\}$.

$$\begin{cases} W^R_{i,j} = max(W^R_{i,j}, W^R_{j,i}), W^R_{j,i} = 0 & W^R_{i,j} \geq W^R_{j,i} \\ W^R_{j,i} = max(W^R_{j,i}, W^R_{i,j}), W^R_{i,j} = 0 & otherwise \end{cases} \tag{9}$$

Finally, the relation of exercise: i with exercise: j is calculated as follows:

$$A_{i,j} = \begin{cases} \mu_1 simi_{i,j} + \mu_2 diff(q_i, q_j) + \mu_3 W^R_{i,j} & if \quad \mu_1 simi_{i,j} + \mu_2 diff(q_i, q_j) + \mu_3 W^R_{i,j} \geq \Theta \\ 0 & otherwise \end{cases} \tag{10}$$

where Θ is a threshold to control the sparsity of the exercise relation matrix. Then Given the past exercises: $(e_1, e_2, \ldots e_{n-1})$ and the next exercise: e_n, the exercise relation matrix is defined as $R^E = [A_{e_n,e_1}, A_{e_n,e_2}, \ldots A_{e_n,e_{n-1}}]$. Finally, the exercise relation matrix is applied as the input of the Position-Relation-Forgetting attention mechanism.

Position-Relation-Forgetting attention mechanism

The Position-Relation-Forgetting attention mechanism includes the relative position attention layer, the relation attention layer, and the forgetting layer. The basic idea of the relative position attention aims to replace the positional encoding of input embedding with relative distance embedding. Specifically, the maximum absolute value of k is used to clip the distance of words in the input embedding. The relative position attention accepts the relative distance between input elements. I_i and I_j serve as the model inputs to track the student state. And edge vectors between I_i and I_j are presented as $a^v_{i,j}$, $a^K_{i,j}$ to extract relative position representations. The edge vectors are clipped to a maximum absolute value of k: clip(x,k) = max(-k, min(k,x)). And corresponding relative position representations are $W^K = (W^k_{-k} \ldots . W^K_k)$ and $W^V = (W^V_{-k} \ldots . W^V_k)$ respectively. The outputs of the relative position attention mechanism are new sequences Z. The process can refer to the following equations:

$$a^K_{i,j} = W^k_{clip(j-i,k)} \quad a^V_{i,j} = W^V_{clip(j-i,k)} \tag{11}$$

$$a_{i,j} = \frac{exp(m_{i,j})}{\sum_{i=1}^{n} exp(m_{i,k})} \quad m_{i,j} = \frac{I_i W^Q (I_j W^k)^T + I_i W^Q (a^K_{i,j})^T}{\sqrt{d_z}} \tag{12}$$

$$Z_i = \sum_{j=1}^{n} a_{ij} (I_j W^V) \tag{13}$$

where W^Q, W^K, and W^V are the query, key, and value matrices respectively and d_z is the dimension of the new sequence of Z. Then, the relation attention predicts student performance on the next interaction by combining the outputs of the relative position attention mechanism.

The relation attention layer incorporates the output of the formula (13) with the exercise relation matrix to predict student performance on the next interaction. The basic idea of the relation attention layer is to incorporate the relation matrix into the attention mechanism to consider the relational information in data.

$$h_i = \frac{E_{e_n} W^Q (Z_j W^K)^T}{\sqrt{d}} \quad \alpha_i = \frac{exp(h_i)}{\sum_{k=0}^{n-1} exp(h_k)} \tag{14}$$

$$\gamma_i = \delta \alpha_i + (1-\delta) R^E_i \quad H = \sum_{i=1}^{n-1} \gamma_i Z_i W^v \tag{15}$$

where W^Q, W^K, and W^V represent the query, key, and value matrices of the attention mechanism and E_{e_n} means the embedding of exercises.

Then applying the output of the relation attention layer is treated as the input of the forgetting layer based on learning theory in the educational field. The basic idea of part is to utilize the time interval between past time and current time to estimate the student forgetting curve. The relative time intervals between past and next interactions are compared as $\triangle_i = t_n - t_i$. The final outputs of the Position-Relation-Forgetting attention mechanism incorporating forgetting behavior, R^F, is computed as follows:

$$R^F = [\xi_1 e^{-\xi_2 \triangle_1}, \xi_1 e^{-\xi_1 \triangle_2} \ldots \xi_1 e^{-\xi_2 \triangle_n}] \quad O = \delta_F H + (1 - \delta_F) R^F \tag{16}$$

where ξ_1 and ξ_2 are hyper-parameters.

Student performance prediction layer

The student performance prediction layer contains the pointwise Feed-Forward (FFN) and probability prediction layer. The FFN can be computed as follows referring to(17). W_l and W_s, b_l and b_s are weighted matrices and bias vectors respectively.

$$F = ReLU(OW_l + b_l)W_s + b_s \tag{17}$$

The probability prediction layer predicts the probability of the student's performance by accepting function: $\sigma()$ on the basis of the FFN. P denotes as the probability that the students provide correct answers in the next interaction. The W and b are trainable parameters.

$$p = \sigma(FW + b) \tag{18}$$

Experiments

Implementation details

Framework setting. The model dimension of attention, the max sequence length, and the training batch size are 200. The dropout rate of the NGFKT model is 1e-2. And the hyper-parameters, including the λ, Θ, are 1 and 0.65 respectively. The parameters in the exercise relation modeling: μ_1, μ_2, and μ_3 are 0.1, 0.2, and 0.7 respectively. The other parameters that are not specified involved in the process of the training model are normally initialized as 0. The details can refer to the following Table 4.

Evaluation metrics

There exist three metrics to measure the performance of our model including the Area Under Curve(AUC), Accuracy(ACC), and Performance Stability(PS). The prediction task is evaluated in a binary classification scenario, i.e., whether or not an exercise is performed correctly. As a result, the AUC and ACC are accepted to measure the prediction performance of students. AUC or ACC values of 0.5 usually indicate that the result was determined at random. The greater the knowledge tracing performance, the higher the value of AUC or ACC. The cross-entropy is accepted as the loss function of the NGFKT model.

Table 4. The framework setting for the Neural Graph Forgetting Knowledge Tracing model(NGFKT).

Parameters	Setting
Max sequence length	200
Training batch size	200
Dropout rate	0.01
λ	1
Θ	0.65
μ_1	0.1
μ_2	0.2
μ_3	0.7

https://doi.org/10.1371/journal.pone.0295808.t004

The Performance Stability metric(PS) is used to specifically compare the performance of the baseline models with the NGFKT model in the testing phase. The performance of the model: M is stable when the M can consistently outperform other models in most testing batches. Based on this idea, the PS is designed based on the performance rank. For instance, if the NGFKT model outperforms the DKT model and DKT+ model in 96 testing batches. However, the performance of the NGFKT model is worse than the DKT+ model in 4 testing batches. The performance rank of the NGFKT model is 1 in 94 testing batches and 2 in 4 testing batches. Then the PS of the NGFKT model is 97.32% referring to the following formulation. The N_{Batch} and N_{model} are the number of the testing batches and models in this paper.

$$PS(M) = \sum_{i=0}^{N_{Batch}} \frac{N_{model} - rank(M, i) + 1}{N_{model}} \tag{19}$$

Datasets

In order to demonstrate the performance of the NGFKT model in the small dataset and the large dataset, two types of public educational datasets were accepted to verify the performance of the NGFKT model in terms of AUC, ACC, and PS.

The statistical information of the datasets is provided in Table 5. There are two public educational datasets to validate the performance of the NGFKT model.

ASSISTments2012(ASSIST2012). The first dataset is the large dataset: ASSISTments2012 (ASSIST2012) (available at: https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect), which was gathered by the *Assistemt Online Tutoring platforms*. The ASSIST2012 contains 4193K records of 39K students. Each student answers an average of 107 questions.

There are several datasets that are collected by *Assistemt Online Tutoring platforms*. The reason for choosing ASSIST2012 is that the NGFKT model needs the timestamp feature in the dataset to model the forgetting behavior of students. However, other datasets *Assistemt Online Tutoring platforms* do not gather the timestamp feature to make predictions. Therefore, ASSIST2012 is accepted as the training dataset to validate the performance of the NGFKT model.

Eedi. The second dataset is the small dataset: the Eedi dataset (available at: https://eedi.com/projects/neurips-education-challenge) collected by the NeuralIPS platform, which includes 233K records and 2064 students. And an average of 113 exercises are provided for each student. In this paper, tasks 3 and 4 in the NeuralIPS system are applied to track the knowledge state of students.

Results and discussion

Student performance prediction. The experimental results are presented in Table 6. The AUC, ACC, and PS are computed to compare the performance of different models. In order to

Table 5. The overview of ASSIST2012 and Eedi based on different intelligent tutoring systems.

Statistic	ASSIST2012	Eedi
Number of records	4193631	233767
Number of students	39364	2064
Number of questions	59761	948
Avg exercise record/student	107	113

https://doi.org/10.1371/journal.pone.0295808.t005

Table 6. Comparison of results of baseline models with the Neural Graph Forgetting Knowledge Tracing model(NGFKT). The NGFKT outperforms all baseline models in terms of AUC, ACC, and PS.

	ASSIST2012			Eedi		
	AUC	ACC	PS	AUC	ACC	PS
DKT	0.712	0.679	0.142	0.489	0.489	0.179
DKT+	0.722	0.685	0.232	0.584	0.566	0.501
DKVMN	0.701	0.686	0.392	0.701	0.640	0.829
SAKT	0.736	0.692	0.732	0.495	0.495	0.216
GKT	0.702	0.687	0.491	0.566	0.542	0.354
NGFKT	**0.776**	**0.704**	**0.960**	**0.710**	**0.673**	**0.937**

https://doi.org/10.1371/journal.pone.0295808.t006

verify the prediction performance of student abilities, the training set and testing set are divided into 80% records of the dataset and 20% records of the dataset respectively.

In two educational datasets, the NGFKT model is compared with five baseline models, including the DKT model, the DKT+ model, the DKVMN model, the SAKT model, and the GKT model. As Table 6 illustrated, the DKT+ model outperforms the DKT model on two datasets due to the fact that two regularization terms are accepted by the DKT+ model to solve the reconstruction problem and the consistency problem in the DKT model. However, knowledge concept modeling is ignored in the DKT model and DKT+ model. Therefore, the DKVMN model applies the nonlinear transformations and student master level on each KC to make predictions and performs better than the DKT+ model and DKT model on the Eedi dataset. However, the DKVMN model performs worse than the DKT model and DKT+ model on the ASSIST2012 because the number of questions is so large on the ASSIST dataset that the Knowledge concepts modeling is excessively complex. Compared with the DKVMN model, the SAKT model further applies the attention mechanism to process the data and discovers the relation between KCs to obtain better performance than the DKVMN model on the ASSIST2012 dataset. However, the SAKT model possesses worse performance than the DKVMN model on the Eedi dataset. The reason is that the number of KCs on the Eedi is less than the ASSIST so the relation modeling of KCs in the SAKT model possesses less effect on the prediction results. In addition, the GKT model does not perform as well as we thought because considering the nodes of the graph established based on two datasets is relatively sparsity. Therefore, the performance of the GKT model is worse than most baseline models. These baseline models do not consider the relation modeling of the exercises and forgetting behavior of students. Therefore, the NGFKT model is proposed to solve these drawbacks. The NGFKT model incorporates the exercise relation modeling with the novel attention mechanism: Position-Relation-Forgetting attention mechanism to make predictions. The NGFKT model performs consistently better than all baseline models in terms of three evaluation metrics.

Ablation experiments. This part aims to find out the key components of our model and validate the performance of these key components. There exist two variants models of the NGFKT model: the NGFKT-PE and the NGFKT-RM, which means that the NGFKT model removes the relative position encoding(PE) or relation modeling(RM)respectively. Specifically, the positional encoding and relation modeling are implemented by the position attention layer and the relation attention layer with the forgetting attention layer. the According to Table 7, some conclusions can be drawn.

Firstly, the individual components of the position attention layer and the relation attention layer do not generate satisfactory outcomes when used alone. The performance of the model is gradually better when more components are involved in the model. Secondly, removing the

Table 7. The ablation experiments of the Neural Graph Forgetting Knowledge Tracing model(NGFKT).

	ASSIST2012			Eedi		
	AUC	ACC	PS	AUC	ACC	PS
NGFKT-RM	0.740	0.678	0.678	0.684	0.637	0.804
NGFKT-PE	0.753	0.689	0.860	0.691	0.641	0.679
NGFKT	**0.776**	**0.704**	**0.960**	**0.710**	**0.673**	**0.937**

https://doi.org/10.1371/journal.pone.0295808.t007

RM component leads to a significant drop, decreasing to 0.740 and 0.684 respectively compared with removing the PE components. Therefore, the RM is a more crucial component for the NGFKT model when the model makes predictions.

Cold start problem. The cold start issue also impacts knowledge tracing when the model predicts the knowledge state of students. The results are examined in two cold start scenarios, i.e., training with data from a small number of students and new students with abbreviated exercise sequences [47].

1. The knowledge tracing model is trained using data from a limited number of students, and it is then applied to completely new, untested samples.
2. When new students enroll in an online learning system, they often have short exercise sequences since there aren’t enough exercise recordings to provide the knowledge tracing model with enough information about them.

In scenario 1, student populations ranging from 10% to 20% of the training dataset are tested to evaluate the prediction performance of NGFKT, DKT, and DKT+ referencing Fig 4. Specifically, the NFGKT model outperforms the DKT model and DKT+ model on the Eedi dataset because the NFGKT model considers the exercise relation modeling and applies the attention mechanism to process the data. And the accuracy of the NGFKT model remains at around 68%. In addition, the performance of the DKT model fluctuates relatively. Compared with the DKT model, the DKT+ model solves the reconstruction problem and the consistency problem to further improve the performance to predict the student knowledge state. The DKT + model’s performance is still at about 53%.

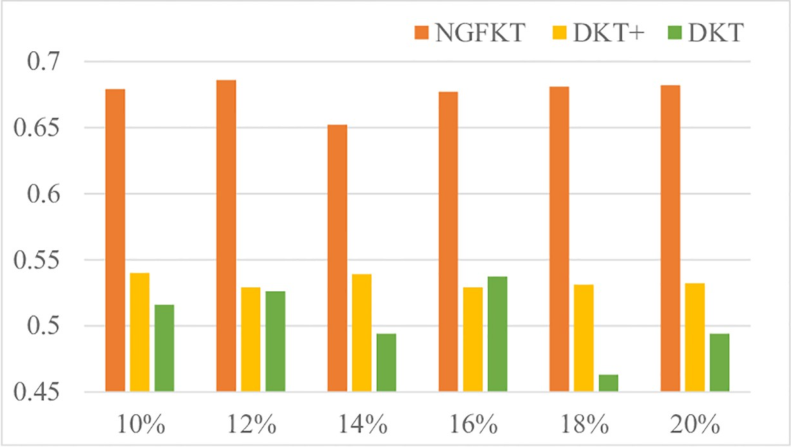

Fig 4. AUC on the Eedi dataset is compared when the number of the students is 10%, 12%, 14%, 16%, 18%, and 20% respectively. The Neural Graph Forgetting Knowledge Tracing Model outperforms the other two models.

https://doi.org/10.1371/journal.pone.0295808.g004

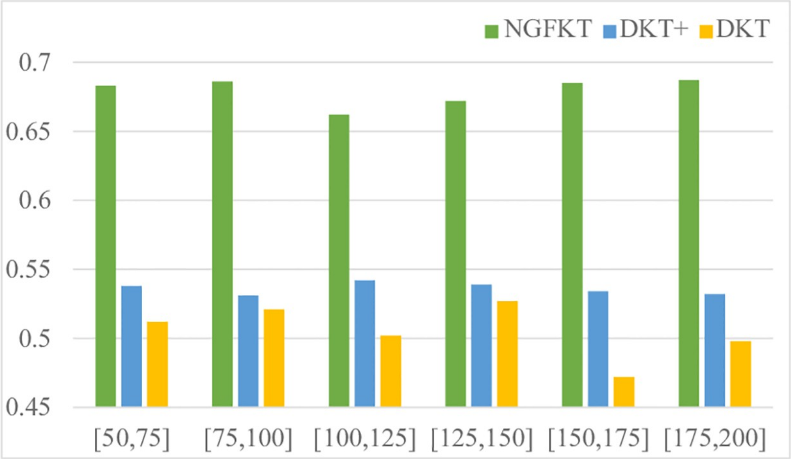

Fig 5. AUC on the Eedi dataset is compared when students are provided with 50, 75, 100, 125, 150, 175 and 200 answer records respectively. The Neural Graph Forgetting Knowledge Tracing Model achieves better results than the other two models.

https://doi.org/10.1371/journal.pone.0295808.g005

In scenario 2, which contains new students who have short exercise sequences, the training data are separated into six groups. Each group has a distinct range of exercise sequences, such as (50, 75], (75, 100], (100, 125], (125, 150], (150, 175], and (175, 200]. The lengths of the exercises from the original exercise sequence are sampled to generate each exercise sequence in the training data. Given that students in the first group have the fewest exercise answering records, it is clear that this situation is the most challenging for the student. In Fig 5, the effectiveness of the various techniques is compared. On the Eedi datasets, the DKT+ model performs better than the DKT model by accepting two regularization items to enhance the model performance. However, the relation modeling part of the DKT+ model is ignored when the DKT+ model predicts the performance of students. Therefore, the NGFKT model solves this problem and shows better performance in this scenario than DKT and DKT+ by considering the relation modeling of the exercises and forgetting behavior of students.

Knowledge state prediction visualization. Knowledge state prediction visualization is regarded as an important application of knowledge tracing models for online educational systems. We will show that our proposed model: the NGFKT model can capture the student knowledge state correctly compared with two standard knowledge tracing models: the DKT model and the DKT+ model. Specifically, Fig 6 indicates the knowledge state traced by the NGFKT model of the same student. The general knowledge evolving process of the knowledge state is consistent with the student learning process. When the student first attempts the exercise, the knowledge state reaches the minimum level. The student continues to learn skills: “32”, “49”, and “71” and continuously deepens his proficiency in knowledge points. Finally, the student knowledge state achieves the maximum, which is shown by the increased areas of the radar diagram. During the latest attempt of the student, the knowledge proficiency of the student presents some reduction considering the student forgetting behavior. However, knowledge proficiency is still improved by continuously practicing the skills compared with the first interaction with the student.

Referring to Fig 6, the NGFKT model also outperforms the DKT model and the DKT + model because the NGFKT model further incorporates the relation modeling that is generated by the GCN model. The DKT+ model achieves better results than the DKT model, which

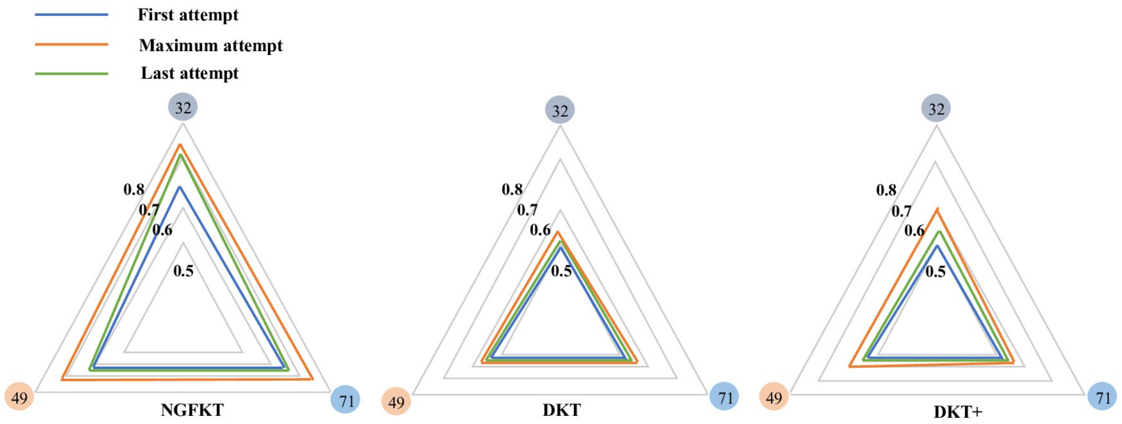

Fig 6. The radar diagram. The NGFKT model outperforms the DKT model and the DKT+ model in tracking the student knowledge state. The "32", "49", and "71" are three skill ids and are presented with three different colors. The average prediction accuracy of the NGFKT model is around 70.5%.

https://doi.org/10.1371/journal.pone.0295808.g006

indicates that adding two regularization terms can further improve the performance of tracing the knowledge proficiency of the student.

Conclusions and future work

In this paper, we introduced a novel knowledge tracing model, NGFKT, designed to accurately track students' knowledge states by integrating relation modeling, skill relation matrices, Q-matrices, and relative distance representations. The NGFKT model effectively predicts students' knowledge levels even with limited interaction data. Our approach employed the KRIRC method to calibrate skill relation matrices and Q-matrices, serving as inputs to a Graph Convolutional Network (GCN) for generating exercise and skill embeddings. By combining skill-exercise embeddings, item difficulty, and the contingency table, we derived the final exercise relation matrix. Employing the Position-Relation-Forgetting Attention mechanism yielded accurate predictions. Our experiments on two public datasets demonstrated the NGFKT model's efficiency in tracking students' knowledge states. The NGFKT model's blend of explanatory and predictive power holds promise for enhancing the design of Online Educational Systems.

In this paper, we properly model the student, exercises, and skills to design an effective knowledge tracing model to estimate the knowledge state of students when considering the heterogeneous graph between them and the exercise difficulty. However, this paper still has two limitations of the knowledge tracing model. Firstly, the skill relation model should take exercise content into consideration to make the prediction more accurate. Secondly, other student behaviors such as guessing behavior or slipping behavior should be further discussed when the student factor is modeled.

In the future, we envision incorporating exercise texts into the knowledge tracing model's design and integrating more nuanced student behaviors—such as the guessing and slipping factors—to enhance performance prediction. Additionally, we plan to expand the knowledge model's scope to develop an innovative online intelligence system for recommending exercises to students. This system aims to create a seamless and tailored learning experience that adapts to individual student's needs and preferences.

Acknowledgments

The authors thank Prof. Chunyan Zeng from Hubei University of Technology for checking the first draft.

Author Contributions

Conceptualization: Linqing Li, Zhifeng Wang.

Data curation: Linqing Li, Zhifeng Wang.

Formal analysis: Linqing Li, Zhifeng Wang.

Funding acquisition: Zhifeng Wang.

Investigation: Linqing Li, Zhifeng Wang.

Methodology: Linqing Li, Zhifeng Wang.

Project administration: Zhifeng Wang.

Supervision: Zhifeng Wang.

Validation: Zhifeng Wang.

Visualization: Zhifeng Wang.

Writing – original draft: Linqing Li, Zhifeng Wang.

Writing – review & editing: Linqing Li, Zhifeng Wang.

References

1. Sun Z, Anbarasan M, Praveen Kumar D. Design of online intelligent English teaching platform based on artificial intelligence techniques. Computational Intelligence. 2021; 37(3):1166–1180. https://doi.org/10.1111/coin.12351
2. Cui Y, Ma Z, Wang L, Yang A, Liu Q, Kong S, et al. A survey on big data-enabled innovative online education systems during the COVID-19 pandemic. Journal of Innovation & Knowledge. 2023; 8 (1):100295. https://doi.org/10.1016/j.jik.2022.100295
3. Ma H, Huang Z, Tang W, Zhang X. Exercise recommendation based on cognitive diagnosis and neutrosophic set. In: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE; 2022. p. 1467–1472.
4. Kang W, Zhang L, Li B, Chen J, Sun X, Feng J. Personalized exercise recommendation via implicit skills. In: Proceedings of the ACM Turing Celebration Conference-China; 2019. p. 1–6.
5. Hamid SNM, Lee TT, Taha H, Rahim NA, Sharif AM. E-content module for Chemistry Massive Open Online Course (MOOC): Development and students' perceptions. JOTSE: Journal of Technology and Science Education. 2021; 11(1):67–92. https://doi.org/10.3926/jotse.1074
6. Idris N, Yusof N, Saad P, et al. Adaptive course sequencing for personalization of learning path using neural network. Int J Advance Soft Comput Appl. 2009; 1(1):49–61.
7. N Bezus S, A Abduzhalilov K, K Raitskaya L. Distance Learning Nowadays: the Usage of Didactic Potential of MOOCs (on platforms Coursera, edX, Universarium) in Higher Education. In: 2020 The 4th International Conference on Education and Multimedia Technology; 2020. p. 14–19.
8. Xia J, Li G, Cao Z. Personalized exercise recommendation algorithm combining learning objective and assignment feedback. Journal of Intelligent & Fuzzy Systems. 2018; 35(3):2965–2973. https://doi.org/10.3233/JIFS-169652
9. de Jong PG, Pickering JD, Hendriks RA, Swinnerton BJ, Goshtasbpour F, Reinders ME. Twelve tips for integrating massive open online course content into classroom teaching. Medical Teacher. 2020; 42 (4):393–397. https://doi.org/10.1080/0142159X.2019.1571569 PMID: 30860930
10. Ai F, Chen Y, Guo Y, Zhao Y, Wang Z, Fu G, et al. Concept-Aware Deep Knowledge Tracing and Exercise Recommendation in an Online Learning System. International Educational Data Mining Society. 2019;.

11. Chen Y, Li X, Liu J, Ying Z. Recommendation system for adaptive learning. Applied psychological measurement. 2018; 42(1):24–41. https://doi.org/10.1177/0146621617697959 PMID: 29335659
12. Tomasevic N, Gvozdenovic N, Vranes S. An overview and comparison of supervised data mining techniques for student exam performance prediction. Computers & education. 2020; 143:103676. https://doi.org/10.1016/j.compedu.2019.103676
13. Yang Y, Shen J, Qu Y, Liu Y, Wang K, Zhu Y, et al. GIKT: a graph-based interaction model for knowledge tracing. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I. Springer; 2021. p. 299–315.
14. Liu S, Zou R, Sun J, Zhang K, Yang J. A Hierarchical Memory Network for Knowledge Tracing. Expert Systems with Applications. 2021; 177(4):114935. https://doi.org/10.1016/j.eswa.2021.114935
15. Jia S, Liu X, Zhao P, Liu C, Sun L, Peng T. Representation of job-skill in artificial intelligence with knowledge graph analysis. In: 2018 IEEE symposium on product compliance engineering-asia (ISPCE-CN). IEEE; 2018. p. 1–6.
16. Piech C, Bassen J, Huang J, Ganguli S, Sahami M, Guibas LJ, et al. Deep knowledge tracing. Advances in neural information processing systems. 2015;28.
17. Zhang J, Shi X, King I, Yeung DY. Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web; 2017. p. 765–774.
18. Nakagawa H, Iwasawa Y, Matsuo Y. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: IEEE/WIC/ACM International Conference on Web Intelligence; 2019. p. 156–163.
19. Pandey S, Karypis G. A self-attentive model for knowledge tracing. arXiv preprint arXiv:190706837. 2019;.
20. Wang Z, Yang Y, Zeng C, Kong S, Feng S, Zhao N. Shallow and Deep Feature Fusion for Digital Audio Tampering Detection. EURASIP Journal on Advances in Signal Processing. 2022; 2022(69):1–20. https://doi.org/10.1186/s13634-022-00900-4
21. Zeng C, Zhu D, Wang Z, Wu M, Xiong W, Zhao N. Spatial and Temporal Learning Representation for End-to-End Recording Device Identification. EURASIP Journal on Advances in Signal Processing. 2021; 2021(1):41. https://doi.org/10.1186/s13634-021-00763-1
22. Mehrish A, Majumder N, Bharadwaj R, Mihalcea R, Poria S. A review of deep learning techniques for speech processing. Information Fusion. 2023; p. 101869. https://doi.org/10.1016/j.inffus.2023.101869
23. Latif S, Zaidi A, Cuayahuitl H, Shamshad F, Shoukat M, Qadir J. Transformers in speech processing: A survey. arXiv preprint arXiv:230311607. 2023;.
24. Wang Z, Wang Z, Zeng C, Yu Y, Wan X. High-Quality Image Compressed Sensing and Reconstruction with Multi-Scale Dilated Convolutional Neural Network. Circuits, Systems, and Signal Processing. 2022; p. 1–24. https://doi.org/10.1007/s00034-022-02181-6
25. Zeng C, Yan K, Wang Z, Yu Y, Xia S, Zhao N. Abs-CAM: A Gradient Optimization Interpretable Approach for Explanation of Convolutional Neural Networks. Signal, Image and Video Processing. 2022; p. 1–8. https://doi.org/10.1007/s11760-022-02313-0
26. Li L, Wang Z, Zhang T. GBH-YOLOv5: Ghost Convolution with BottleneckCSP and Tiny Target Prediction Head Incorporating YOLOv5 for PV Panel Defect Detection. Electronics. 2023; 12(3):1–15. https://doi.org/10.3390/electronics12030561
27. Zeng C, Ye J, Wang Z, Zhao N, Wu M. Cascade Neural Network-Based Joint Sampling and Reconstruction for Image Compressed Sensing. Signal, Image and Video Processing. 2022; 16(1):47–54. https://doi.org/10.1007/s11760-021-01955-w
28. Xiong X, Zhao S, Van Inwegen EG, Beck JE. Going deeper with deep knowledge tracing. International Educational Data Mining Society. 2016;.
29. Liu Q, Huang Z, Yin Y, Chen E, Xiong H, Su Y, et al. Ekt: Exercise-aware knowledge tracing for student performance prediction. IEEE Transactions on Knowledge and Data Engineering. 2019; 33(1):100–115. https://doi.org/10.1109/TKDE.2019.2924374
30. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T. Meta-learning with memory-augmented neural networks. In: International conference on machine learning. PMLR; 2016. p. 1842–1850.
31. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:180601261. 2018;.
32. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907. 2016;.
33. Chanaa A, Faddouli E, et al. Predicting learners need for recommendation using dynamic graph-based knowledge tracing. In: International conference on artificial intelligence in education. Springer; 2020. p. 49–53.

34. Song X, Li J, Sun S, Yin H, Dawson P, Doss RRM. SEPN: a sequential engagement based academic performance prediction model. IEEE Intelligent Systems. 2020; 36(1):46–53. https://doi.org/10.1109/MIS.2020.3006961

35. Wu Z, Huang L, Huang Q, Huang C, Tang Y. SGKT: Session graph-based knowledge tracing for student performance prediction. Expert Systems with Applications. 2022; 206:117681. https://doi.org/10.1016/j.eswa.2022.117681

36. Nagatani K, Zhang Q, Sato M, Chen YY, Chen F, Ohkuma T. Augmenting knowledge tracing by considering forgetting behavior. In: The world wide web conference; 2019. p. 3101–3107.

37. Gan W, Sun Y, Sun Y. Knowledge structure enhanced graph representation learning model for attentive knowledge tracing. International Journal of Intelligent Systems. 2022; 37(3):2012–2045. https://doi.org/10.1002/int.22763

38. Pandey S, Srivastava J. RKT: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management; 2020. p. 1205–1214.

39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. 2017;30.

40. Nayakanti N, Al-Rfou R, Zhou A, Goel K, Refaat KS, Sapp B. Wayformer: Motion forecasting via simple & efficient attention networks. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2023. p. 2980–2987.

41. Joshi A, Fidalgo E, Alegre E, Fernández-Robles L. DeepSumm: Exploiting topic models and sequence to sequence networks for extractive text summarization. Expert Systems with Applications. 2023; 211:118442. https://doi.org/10.1016/j.eswa.2022.118442

42. Vardasbi A, Pires TP, Schmidt RM, Peitz S. State Spaces Aren't Enough: Machine Translation Needs Attention. arXiv preprint arXiv:230412776. 2023;.

43. Kang L, He S, Wang M, Long F, Su J. Bilingual attention based neural machine translation. Applied Intelligence. 2023; 53(4):4302–4315. https://doi.org/10.1007/s10489-022-03563-8

44. Wu M, Foster G, Qu L, Haffari G. Document flattening: Beyond concatenating context for document-level neural machine translation. arXiv preprint arXiv:230208079. 2023;.

45. Ranathunga S, Lee ESA, Prifti Skenduli M, Shekhar R, Alam M, Kaur R. Neural machine translation for low-resource languages: A survey. ACM Computing Surveys. 2023; 55(11):1–37. https://doi.org/10.1145/3567592

46. Rei R, Guerreiro NM, Treviso M, Coheur L, Lavie A, Martins AF. The inside story: Towards better understanding of machine translation neural evaluation metrics. arXiv preprint arXiv:230511806. 2023;.

47. Zhao J, Bhatt S, Thille C, Gattani N, Zimmaro D. Cold Start Knowledge Tracing with Attentive Neural Turing Machine. In: Learning at Scale; 2020.